ALTAS
MANUAL DE USUARIO
SERVICIO EREG

Versión 1.2

Unidad de Arquitectura del Software

Agencia de Informática y Comunicaciones de la Comunidad de Madrid

Hoja de Control

Título	Manual de Usuario Servicio de EREG		
Documento de Referencia	NORMATIVA ATLAS		
Responsable	Unidad de Arquitectura del Software		
Versión	1.2	**Fecha Versión**	30/04/2014

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Unidad de Arquitectura del Software	09/10/2013
1.1	Inclusión EregConsultaService y EregTramitacionService	Unidad de Arquitectura del Software	24/10/2013
1.2	Inclusión de métodos AnexarFichAnexoDocum y AnexarFichSolicitudDocum	Unidad de Arquitectura del Software	03/01/2014

Índice

1. INTRODUCCIÓN 4

1.1. AUDIENCIA OBJETIVO 4
1.2. CONOCIMIENTOS PREVIOS 4

2. DESCRIPCIÓN 5

3. INSTALACIÓN Y CONFIGURACIÓN 6

3.1. INSTALACIÓN 6
3.1.1. Paso 1: Añadir la dependencia al módulo de Administración Electrónica 6
3.1.2. Paso 2: Incluir la ruta de definición del servicio 'EregAltaService' 7
3.1.3. Paso 3: Configurar el endpoint del servicio 7
3.1.4. Paso 4: Inyectar el Servicio en las clases que lo requieran 7

4. USO DEL SERVICIO EREGALTASERVICE 9

4.1. MÉTODO REGISTRARENTRADA 9
4.2. MÉTODO ALTADOCUM 11
4.3. MÉTODO ANEXARFICHANEXODOCUM 12
4.4. MÉTODO ANEXARFICHSOLICITUDDOCUM 13
4.5. MÉTODO REGISTRARSALIDA 14
4.6. MÉTODO GETRELANEXOSUNIFICADO 16
4.7. MÉTODO ASOCIAR_INTERESADO 17
4.8. MÉTODO LOADANEXODOCUM 18
4.9. MÉTODO ANULARDOCUMENTO 19
4.10. MÉTODO GETFORMULARIO 19
4.11. MÉTODO GENERARPRESALIDA 20

5. USO DEL SERVICIO EREGCONSULTASERVICE 21

5.1. MÉTODO DOWNLOADANEXODOCUM 21
5.2. MÉTODO GETANOREGDOCUM 21
5.3. MÉTODO GETASUNUNIDAD 22
5.4. MÉTODO GETDATOSGENDOCUM 23
5.5. MÉTODO GETINFODOCUM 24
5.6. MÉTODO GETXMLANEXO 24
5.7. MÉTODO GETXMLSOLICITUD 24
5.8. MÉTODO LOADSOLICITUDDOCUM 25

6. USO DEL SERVICIO EREGTRAMITACIONSERVICE 26

6.1. MÉTODO BANDEJAENTRADA4 26
6.2. MÉTODO ENVIARDOCUM 28
6.3. MÉTODO RECEPCIONARDOCUM 28
6.4. MÉTODO VALVINCDOCUM 28

7. ENLACES RELACIONADOS 30

Contenido

1. INTRODUCCIÓN

Este documento contiene instrucciones y ejemplos de uso de la librería de servicios de EREG.. Para más información sobre las capacidades funcionales de este servicio, debe consultarse la documentación original del proyecto.

La librería se compone de 3 servicios:

- **EregAltaService** – Servicios de administración electrónica relacionados con el Alta
- **EregConsultaService** - Servicios de administración electrónica relacionados con la Consulta
- **EregTramitacionService** - Servicios de administración electrónica relacionados con la Tramitación

1.1. Audiencia objetivo

El lector objetivo de este documento es toda aquella persona que esté desarrollando una solución en la que se necesite usar servicios de EREG en sus aplicación ATLAS.

1.2. Conocimientos Previos

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Lenguaje Java
- Spring Framework

2. DESCRIPCIÓN

El servicio de Spring "**EregAltaService**" que incluye el módulo de administración electrónica de Atlas, provee las siguientes funcionalidades, que se explican en los siguientes apartados:

- altaDocum
- AnexarFichAnexoDocum
- AnexarFichSolicitudDocum
- anularDocumento
- asociar_Interesado
- generarPresalida
- getFormulario
- getRelAnexosUnificado
- loadAnexoDocum
- registrarEntrada
- registrarSalida

El servicio de Spring "**EregConsultaService**" que incluye el módulo de administración electrónica de Atlas, provee las siguientes funcionalidades, que se explican en los siguientes apartados:

- downloadAnexoDocum
- getAnoregDocum
- getAsunUnidad
- getDatosGenDocum
- getInfoDocum
- getXMLAnexo
- getXMLSolicitud
- loadSolicitudDocum

El servicio de Spring "**EregTramitacionService**" que incluye el módulo de administración electrónica de Atlas, provee las siguientes funcionalidades, que se explican en los siguientes apartados:

- bandejaEntrada4
- enviarDocum
- recepcionarDocum
- valVincDocum

3. INSTALACIÓN Y CONFIGURACIÓN

3.1. INSTALACIÓN

El Servicio '***EregAltaService'*** está incluido en la librería ***atlasfrm-eadm-lib***. Para poder utilizarlo, bastará con incluir la dependencia de este módulo en el fichero *pom.xml* del proyecto, añadir la ruta correspondiente al fichero *xml* de Spring en la variable '*configLocations'*, y definir en el contexto de Spring la configuración del servicio para la aplicación., según se describe en los siguientes apartados:

3.1.1. Paso 1: Añadir la dependencia al módulo de Administración Electrónica

Atención
Este paso no es necesario si el proyecto se ha generado a partir del arquetipo *web, servicioweb* o *documentumweb* de Atlas, ya que dicha librería ya viene incluida por defecto en dichos arquetipos.

Para añadir la dependencia al módulo de seguridad dentro de nuestro proyecto *maven* hay que añadir una entrada "**dependency**" en la sección "**dependencies**" del fichero "**pom.xml**" del proyecto, como se puede ver en el siguiente ejemplo:

pom.xml

```
<dependencies>
      [...]
      <dependency>
          <groupId>atlasfrm</groupId>
          <artifactId>atlasfrm-eadm-lib</artifactId>
          <version>${atlasfrm-eadm-lib.version}</version>
      </dependency>
      [...]
</dependencies>
```

3.1.2. Paso 2: Incluir la ruta de definición del servicio 'EregAltaService'

La librería de administración electrónica ya contiene una definición del servicio '**EregAltaService**' dentro del contexto de Spring. Bastará con incluir la siguiente ruta en una sentencia <import> dentro de un fichero de configuración de Spring (por ejemplo, en el fichero applicationContext-general.xml):

applicationContext-general.xml

```
<import resource="classpath:/conf/applicationContext-eadm-lib.xml"/>
```

3.1.3. Paso 3: Configurar el endpoint del servicio

El enpoint del servicio está definido en las variables por defecto de ATLAS.

Para incluir las variables por defecto de ATLAS comprobar el fichero `applicattionContext-general.xml` para ver si está incluido la lectura del fichero `common.properties`. Si la línea en amarillo no está incluida hay que incluirla.

applicationContext-general.xml

```
    <bean id="propertyConfigurer"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="locations">
            <list>
                <value>classpath:/conf/common.properties</value>
                <value>classpath:/conf/application.properties</value>
                <value>classpath:/environment.properties</value>
                <value>classpath:/conf/queries.properties</value>
            </list>
        </property>
    </bean>
```

3.1.4. Paso 4: Inyectar el Servicio en las clases que lo requieran

Para inyectar el Servicio previamente definido, se pasa como "**property**" al bean que lo vaya a usar, que debe tener el método "**set**" correspondiente. Para más información sobre cómo inyectar referencias, se puede consultar la documentación de Spring.

applicationContext-services.xml

```
<bean id="miService" class="xxxx.services.MiServiceImpl"
      p:eregAltaService-ref="eregAltaService"/>
```

xxxx.services.MiServiceImpl.java

```
public class MiServiceImpl implements MiService {

    private EregAltaService eregAltaService;

    public void setEregAltaService(EregAltaService eregAltaService) {
        this.eregAltaService = eregAltaService;
    }
    ...
}
```

4. USO DEL SERVICIO EregAltaService

Una vez configurado el entorno para el uso del servicio, puede procederse a su utilización.

4.1. Método registrarEntrada

EregAltaService.java

```
/**
 * Servicio para permitir realizar el Registro de Entrada en la Comunidad de
 * Madrid, de solicitudes, ofertas o documentos, desde sistemas de información
 * externos a EREG.
 * Genera un asiento registral, un registro de seguimiento del asiento
 * registral y un numero de documento.
 * Es obligatorio informar el NIF del interesado y Nombre y 1 Apellido o NIF
 * del interesado y Razón Social.
 * Es obligatorio informar el código del asunto, si este no tuviera una
 * unidad asociada es decir puede tener varias unidades asociadas, por
 * ejemplo el código de asunto "FACTURA" no tiene asociada ninguna unidad,
 * en este caso se debe informar el Código de Unidad.
 *
 * @param datosEntrada Objeto {@link EregRegistrarEntradaIN}
 * @return Objeto de tipo {@link EregRegistrarEntradaOUT}
 * @throws ServiceException
 */
 EregRegistrarEntradaOUT registrarEntrada(EregRegistrarEntradaIN datosEntrada)
            throws ServiceException;
```

Objeto con los parámetros de entrada del método registrarEntrada:

EregRegistrarEntradaIN.java

```
/**
 * Objeto con los datos de entrada para el método registrarEntrada
 *
 * cdaplicacion: Código de la aplicación varchar2(5)
 * fcregent: Fecha y hora de registro de entrada
 * cdorigenanoreg: Origen Anotación Registral: P = Presencial, I = Internet
 * cod_asunto: Código del asunto varchar2(16)
 * cdunidaddest: Código unidad destino varchar2(16) (opcional)
 * interNifcif: NIF/CIF del interesado varchar2(17)
 * interNombre: Nombre del interesado varchar2(30) (opcional)
 * interApellido1: Primer Apellido del interesado varchar2(25) (opcional)
 * interApellido2: Segundo Apellido del interesado varchar2(25) (opcional)
 * interRazonsocial: Razón Social del interesado varchar2(80) (opcional)
 * cdprov: Código provincia varchar2(2) (opcional)
 * cdmuni: Código municipio varchar2(3) (opcional)
 * preNifcif: NIF/CIF del presentador varchar2(17)
 * preNombre: Nombre del presentador varchar2(30)
 * preApellido1: Primer apellido del presentador varchar2(25)
 * preApellido2: Segundo apellido del presentador varchar2(25)
 * claseCert: Clase del Certificado varchar2(20) (opcional)
 * numCert: Número de Certificado varchar2(10) (opcional)
 * estadodocumento: Estado del documento 3 = enviado, 4 = recibido,
 *                  23 = Bandeja unidad y enviado, 24 = Bandeja unidad y recibido
 * iddocumento_padre: Identificador documento padre varchar2(16) (opcional)
 * cdexpdte: Numero referencia Expediente varchar2(25) (opcional)
 * campoaux1: Campo auxiliar 1 (opcional)
 * campoaux2: Campo auxiliar 2 (opcional)
 */
public class EregRegistrarEntradaIN;
```

Objeto con los parámetros de salida del método registrarEntrada:

EregRegistrarEntradaOUT.java

```
/**
 * Objeto con los datos de salida para el método registrarEntrada
 *
 * cddocum: Código de Documento varchar2(11)
 * cdstatus: 0 = ok, distinto de 0 = error varchar2(1)
 * dsrefdocum: Nº Referencia del Documento varchar2(14)
 * dsstatus: Descripción en caso de error varchar2(100)
 * dsunidaddest: Descripción Unidad de Destino del Documento varchar2(100)
 * fcpres: Fecha y hora de registro de Entrada.
 * numreg: Número de asiento registral varchar2(5)
 */
public class EregRegistrarEntradaOUT;
```

4.2. Método altaDocum

EregAltaService.java

```
/**
 * Servicio para dar de alta un documento.
 *
 * @param datosEntrada Objeto {@link EregAltaDocumIN} con los parámetros de entrada
 * @return Código del documento insertado varchar2(11)
 * @throws ServiceException
 */
public String altaDocum(EregAltaDocumIN datosEntrada) throws ServiceException;
```

Objeto con los datos de entrada del método altaDocum

EregAltaDocumIN.java

```
/**
 * Objeto con los datos de entrada para el método altaDocum
 *
 * cdUsuario Login del Usuario varchar2(10)
 * fcCrea Fecha de creación del documento Formato: 28/11/2006 13:21:00 varchar2(19)
 * cdCenGestor Centro Gestor varchar2(10)
 * cdOrgaDest Código unidad Administrativa. destino del documento varchar2(16)
 * dsDocum Descripción del documento varchar2(240)
 * cdOrgaCrea Unidad administrativa creadora del documento varchar2(16)
 * dsObserv Texto libre de observaciones varchar2(240) (opcional)
 * cdDocumPadre Código del documento padre varchar2(11) (opcional)
 */
public class EregAltaDocumIN;
```

4.3. Método AnexarFichAnexoDocum

EregAltaService.java

```
    /**
     * Asocia a la anotación registral que se pasa en el campo cddocum el anexo
     * informado en el campo fichero. Sera necesario especificar cada uno
     * de los parámetros del método. El campo cdmimetype será la extensión del
     * fichero anexo. El tipo de ficheros que se podrán adjuntar serán los de
     * Ofimática de Microsoft (doc, rtf, ppt, xls, docx, xlsx, pptx, mdbx, mdb) y
     * los genéricos pdf, tif, jpg, xml, txt, zip, el tamaño del fichero adjunto no
     * debe superar los 4 Megabytes.
     *
     * El campo hashFirma será un hash del fichero subido, firmado con el
     * tipo de firma que se indica
     *
     * @param datosEntrada Objeto {@link EregFichAnexoDocumIN}
     * @return nmanexo Códigos del anexo y de la firma en el repositorio documental
     *         separados por una coma
     * @throws ServiceException
     */
    public String AnexarFichAnexoDocum(EregFichAnexoDocumIN datosEntrada)
            throws ServiceException;
```

Objeto con los datos de entrada del método AnexarFichAnexoDocum

EregFichAnexoDocumIN.java

```
/**
 * Objeto con los datos de entrada para el método AnexarFichAnexoDocum de
 * @atlas.eadm.services.EregAltaService#AnexarFichAnexoDocum(EregFichAnexoDocumIN)
 *  cdModuloTec Módulo técnico de la aplicación que invoca el método
 *  cdDocum Número de referencia de la anotación registral a la que se asociará el
 *          fichero
 *  dsFichero Descripción del fichero anexo
 *  cdMimeType Extensión del fichero anexo
 *  fichero InputStream con fichero a anexar
 *  firma InputStream con la firma del fichero adjunto. (Si la firma es XADES se
 *        firmará el hash del fichero)
 *  tipoFirma tipo de firma con el que se ha firmado el Hash
*/

public class EregFichAnexoDocumIN;
```

4.4. Método AnexarFichSolicitudDocum

EregAltaService.java

```
    /**
     * Asocia a la anotación registral que se pasa en el campo Cddocum el
     * documento informado en el campo FciheroBase64. Este fichero deberá ser
     * uno de los ficheros que conforman una solicitud telemática, clasificados
     * según se puede ver en los parámetros de entrada.
     *
     * En el caso de que, para un mismo Cddocum, en diferentes llamadas a este
     * método, se incluya un mismo tipo de fichero, se obtendrá un mensaje de
     * error indicando que esa solicitud telemática ya tiene informado ese tipo
     * de fichero.
     *
     * @param datosEntrada Objeto {@link EregFichSolicitudDocumIN} con los
     *        parámetros de entrada
     * @return nmanexo Código del fichero en el repositorio documental
     * @throws ServiceException
     */
    public String AnexarFichSolicitudDocum(EregFichSolicitudDocumIN datosEntrada)
            throws ServiceException;
```

Objeto con los datos de entrada del método AnexarFichSolicitudDocum

EregFichSolicitudDocumIN.java

```
/**
 * Objeto con los datos de entrada para el método AnexarFichSolicitudDocum de
 * atlas.eadm.services.EregAltaService#AnexarFichSolicitudDocum(…)
 *
 * cdModuloTec Módulo técnico de la aplicación que invoca el método</li>
 * cdDocum  Número de referencia de la anotación registral a la que se asociará
 *          el fichero
 * cdTipoFichero Podrá tener los siguientes valores:
 *     T1 - PDF original Firmado con PADES
 *     T2 - PDF del justificante de la solicitud
 *     T3 - FDF
 *     T4 - XFDF del T1
 *     T5 - XFDF del T2
 * fichero  InputStream con fichero a anexar
 */

public class EregFichSolicitudDocumIN;
```

4.5. Método registrarSalida

EregAltaService.java

```
/**
 * Servicio para permitir realizar el Registro de Salida de documentos asociando un
 * único interesado
 *
 * @param datosEntrada Objeto de tipo {@link EregRegistrarSalidaIN}
 * @return Objeto de tipo {@link EregRegistrarSalidaOUT}
 * @throws ServiceException
 */
public EregRegistrarSalidaOUT registrarSalida(EregRegistrarSalidaIN datosEntrada)
            throws ServiceException;
```

Objeto con los parámetros de entrada del método registrarSalida

EregRegistrarSalidaIN.java

```
/**
 * Objeto con los datos de entrada para el método registrarSalida
 *
 * cdaplicacion Código de la aplicación varchar2(10)
 * cdusuario Login del usuario varchar2(10)
 * cddocum Código del documento varchar2(11) (opcional)
 * cdasunto Código del asunto varchar2(16)
 * cdunidadremi Código de la unidad remitente varchar2(16)
 * cdunidaddest_int Código de la unidad destino interna varchar2(16)(opcional)
 * cdunidaddes_ext Código de la unidad destino externa varchar2(16) (opcional)
 * nombre_interesado Nombre del interesado varchar2(30) (opcional)
 * apellido1 Primer Apellido del interesado varchar2(25) (opcional)
 * apellido2 Segundo Apellido del interesado varchar2(25) (opcional)
 * razon_social Razón social del interesado varchar2(80) (opcional)
 * cod_doc_identificacion Doc. identificativo del interesado varchar2(17) (Opcional)
 * pais País del interesado varchar2(3) (opcional)
 * cod_provincia Código de la provincia del interesado varchar2(2)(opcional)
 * cod_municipio Código del municipio del interesado varchar2(3) (opcional)
 * calle Calle del interesado varchar2(100) (opcional)
 * direccion Ampliación/aclaración del campo calle varchar2(100)(opcional)
 * cod_postal Código postal del interesado varchar2(5) (opcional)
 * it_acuse Acuse de recibo varchar2(1)
 * campoaux1 Campo auxiliar 1 (opcional)
 * campoaux2 Campo auxiliar 1 (opcional)
 */
  public class EregRegistrarSalidaIN;
```

Objeto con los parámetros de salida del método registrarSalida

EregRegistrarSalidaOUT.java

```
/**
 *  Objeto con los datos de salida para el método registrarSalida
 *
 * cod_documento        Código del documento
 * ds_ref_documento     Referencia del documento - Ej: 99/000251.9/10
 * id_anotacion_reg     Identificador de la anotación registral
 * num_registral        Entidad registral
 * fecha_registro_sal   Fecha registro salida - Ej: 11/02/2010 14:26
 * cod_resultado        Código resultado inserción anotación
 * ds_resultado         Descripción resultado inserción anotación
 * interesados          Array de tipo {@link EregInteredado}
 */
  public class EregRegistrarSalidaOUT;
```

Objeto EregInteresado para la lista de interesados de los parámetros de salida del método registrarSalida

EregInteresado.java

```
/**
 * Objeto interesado para el array de interesados de
 * {@link EregRegistrarSalidaOUT} que se usa en los datos del salida del método
 * registrarSalida
 *
 * id_interesado Identificador del interesado
 * cod_resultado Código resultado inserción interesado
 * ds_resultado Descripción resultado inserción interesado
 */
 public class EregInteresado;
```

4.6. Método getRelAnexosUnificado

EregAltaService.java

```
/**
 * getRelAnexosUnificado Servicio para obtener todos los archivos anexos a
 * un documento, sea cual sea su tipología
 *
 * @param cdDocum Número de referencia de documento válido de registro
 * @return Objeto {@link EregRelAnexosUnificadoOUT} con los anexos
 * @throws ServiceException
 */
public EregRelAnexosUnificadoOUT getRelAnexosUnificado(String cdDocum)
     throws ServiceException;
```

Objeto con los parámetros de salida del método getRelAnexosUnificado

EregRelAnexosUnificadoOUT.java

```
/**
 * Objeto con los datos de salida para el método getRelAnexosUnificado
 * documento_ref_doc
 * idDocumento
 * Anexos   Array de anexos {@link EregAnexo} con:
 *        cod_archivo
 *        descripcion
 *        tipo_archivo
 *        cod_aplicacion
 *        fecha_alta
 */
public class EregRelAnexosUnificadoOUT;
```

Objeto anexo para los parámetros de salida del método getRelAnexosUnificado

EregAnexo.java

```
/**
 * Objeto anexo para el array de anexos de {@link EregRelAnexosUnificadoOUT} que se
 * usa en los datos del salida del métodogetRelAnexosUnificado
 *
 *  cod_archivo
 *  descripcion
 *  tipo_archivo
 *  cod_aplicacion
 *  fecha_alta
 */
 public class EregAnexo;
```

4.7. Método asociar_Interesado

EregAltaService.java

```
/**
  * Servicio que permite asociar un interesado a un documento. Como respuesta
  * devuelve el código y la descripción del Asunto, la fecha y hora de
  * registro de entrada, indicando si ha sido enviado telemáticamente o no.
  *
  * @param datosEntrada Objeto de tipo {@link EregAsociarInteresadoIN}
  * @return String (plano) con la respuesta
  *         Ejemplo: El Apunte registral: 168664 no corresponde al documento:
  *         03050097039
  * @throws ServiceException
  */
 public String asociar_Interesado(EregAsociarInteresadoIN datosEntrada)
     throws ServiceException;
```

Objeto con los parámetros de entrada del método asociar_Interedado

EregAsociarInteresadoIN.java

```
/**
 * Objeto con los datos de entrada para el método asociar_Interesado
 *
 *  cdUsuario Login del Usuario varchar2(10)
 *  cdDocumento Código del documento varchar2(11)
 *  idAnoreg Identificador de la anotación registral varchar2(16)
 *  nombreInteresado Nombre del interesado varchar2(30) (Opcional)
 *  apellido1 Apellido1 del interesado varchar2(25) (Opcional)
 *  apellido2 Apellido2 del interesado varchar2(25) (Opcional)
 *  razonSocial Razon Social varchar2(80) (Opcional)
 *  cdPais Código de País varchar2(3) (Opcional)
 *  tipoDocIdentificacion Tipo de Interesado varchar2(1)
 *  codDocIdentificacion Código de Interesado varchar2(9)
 *  codProvincia Código de Provincia varchar2(3) (Opcional)
 *  codMunicipio Código de Municipio varchar2(3) (Opcional)
 *  Calle Calle varchar2(100) (Opcional)
 *  direccion Dirección varchar2(200) (Opcional)
 *  codigoPostal Código Postal varchar2(5) (Opcional)
 *  itAcuse Acuse de recibo S ó N varchar2(1)
 */
public class EregAsociarInteresadoIN;
```

4.8. Método loadAnexoDocum

EregAltaService.java

```
/**
 * Servicio para anexar un fichero a un documento
 *
 * @param datosEntrada Objeto de tipo {@link EregAnexoDocumIN}
 * @return Identificador del anexo
 * @throws ServiceException
 */

 public String loadAnexoDocum(EregAnexoDocumIN datosEntrada)
        throws ServiceException;
```

Objeto con los parámetros de entrada del método loadAnexoDocum

EregAnexoDocumIN.java

```
/**
* Objeto con los datos de entrada para el método loadAnexoDocum
*
* cddocum Código documento varchar2(11)
* anexo InputStream con el anexo
* descripcion Descripción del anexo varchar2(100)
* dmimetype Tipo de anexo: doc, rtf, ppt, xls, pdf, tif, jpg, xml
**/
  public class EregAnexoDocumIN;
```

4.9. Método anularDocumento

EregAltaService.java

```
/**
 * Servicio que permite anular un número de documento del registro
 *
 * @param codUsuario Login del usuario
 * @param numRef Número de referencia del documento
 * @return Objeto {@link EregAnularDocumentoOUT} con los parámetros de salida
 * @throws ServiceException
 */
public EregAnularDocumentoOUT anularDocumento(String codUsuario, String numRef)
            throws ServiceException;
```

Objeto con los parámetros de salida del método anularDocumento

EregAnularDocumentoOUT.java

```
/**
 * Objeto con los datos de salida para el método anularDocumento
 *
 * cdError Código de error. 0 si no hay error, -1 si hay error
 * dsError Descripcion del error
*/

public class EregAnularDocumentoOUT:
```

4.10. Método getFormulario

EregAltaService.java

```
/**
 * Servicio para obtener el formulario a partir de su código impreso.
 *
 * @param cdimpreso: Código de impreso
 * @return Un array de bytes con el fichero
 * @throws ServiceException
 */

    public byte[] getFormulario(String cdimpreso) throws ServiceException;
```

4.11. Método generarPresalida

EregAltaService.java

```
/**
 * Servicio para generar la presalida
 *
 * @param datosEntrada Objeto {@link EregGenerarPresalidaIN}
 * @throws ServiceException
 */
    public void generarPresalida(EregGenerarPresalidaIN datosEntrada)
          throws ServiceException;
```

Objeto con los parámetros de entrada de generarPresalida

EregGenerarPresalidaIN.java

```
/**
* Objeto con los datos de entrada para el método generarPresalida
*
* cdUsuario Login del Usuario varchar2(10)
* cdDocum Código del documento varchar2(11)
* cdUnidadDest Código Unidad Admtiva de destino varchar2(16) (Opcional)
* tipDoc Tipo de documento de identificación: Nif: N - Cif: C - Nie: E - UT: U
* docIdent Número de documento de identificación   varchar2(9)
* nombre varchar2(30) (Opcional)
* apellido1 varchar2(25) (Opcional)
* apellido2 varchar2(25) (Opcional)
* razSoc Razón social varchar2(80)  (Opcional)
* pais varchar2(3) (Opcional)
* provincia varchar2(2) (Opcional)
* localidad varchar2(3) (Opcional)
* calle varchar2(100) (Opcional)
* direccion varchar2(200) (Opcional)
* cp Código Postal varchar2(5) (Opcional)
* telf Teléfono varchar2(13) (Opcional)
* fax varchar2(13) (Opcional)
* email varchar2(50) (Opcional)
* itAcuse Acuse de Recibo (S/N) varchar2(1)
*/

public class EregGenerarPresalidaIN;
```

5. USO DEL SERVICIO EregConsultaService

5.1. Método downloadAnexoDocum

EregConsultaService.java

```
/**
 * Servicio para obtener el anexo de un documento
 *
 * @param cdDocum Código del documento
 * @param nmAnexo Número de anexo
 * @return Un array de bytes con el fichero
 * @throws ServiceException
 */
public byte[] downloadAnexoDocum(String cdDocum, String nmAnexo)
        throws ServiceException;
```

5.2. Método getAnoregDocum

EregConsultaService.java

```
/**
 * Servicio para obtener las anotaciones registrales a las cuales está
 * asociado un documento
 * @param cdDocum Código de la Solicitud
 * @return Objeto de tipo {@link EregAnotacionesRegistralesOUT}
 * @throws ServiceException
 *
 */
public EregAnotacionesRegistralesOUT getAnoregDocum(String cdDocum)
        throws ServiceException;
```

Objeto con los datos de salida del método

EregAnotacionesRegistralesOUT.java

```
/**
 * Objeto con los datos de salida para el método getAnoregDocum de
 * {@link atlas.eadm.services.EregConsultaService#getAnoregDocum(String)}
 * Contiene una lista de anotaciones registrales {@link EregAnotacionRegistral}
 */

public class EregAnotacionesRegistralesOUT;
```

Objeto anotación registral

EregAnotacionRegistral.java

```
/**
 * Objeto Anotación Registal que se usa en {@link EregAnotacionesRegistralesOUT}
 * que es el objeto de salida
 *
 * Campos:
 * idanoreg: Identificador de la anotación
 * cdtipoanoreg: Código del tipo de anotación registral
 * dstipoanoreg: Descripción del tipo de anotación registral
 * fcanoreg: Fecha de registro. Formato: 12/02/2007 12:10:02
 * cdoriganoreg: Código del origen de la anotación
 * dsoriganoreg: Descripción del origen de la anotación<7li>
 * nmanoreg: Número de la anotación
 * anoreg: Año de la anotación
 * dsasunto: Descripción del asunto
 * idregent: Identificador del registro de entrada
 * dsregent: Descripción del registro de entrada
 * idofireg: Identificador de la oficina de registro
 * dsofireg: Descripción de la oficina de registro
 *
 */
public class EregAnotacionRegistral;
```

5.3. Método getAsunUnidad

EregConsultaService.java

```
    /**
     * Servicio para obtener la relación de códigos y descripciones de asuntos
     * asociados a una Unidad Administrativa
     *
     * @param cdUnidad Código Unidad Administrativa
     * @return un Map<String, String> con todos los pares de código
     * asunto=descripción asunto
     * @throws ServiceException
     */
    public Map<String, String> getAsunUnidad(String cdUnidad)
            throws ServiceException;
```

5.4. Método getDatosGenDocum

EregConsultaService.java

```
    /**
     *
     * Servicio para obtener los datos generales de un Documento.
     * @param cdDocum Código de la Solicitud
     * @return Objeto de tipo {@link EregDatosGenDocumOUT} con los datos de salida
     * @throws ServiceException
     */
    public EregDatosGenDocumOUT getDatosGenDocum(String cdDocum)
         throws ServiceException;
```

Objeto con los datos de salida

EregDatosGenDocumOUT.java

```
/**
 * Objeto con los datos de salida para el método getDatosGenDocum de
 *
 *   cddocum: Código de la solicitud
 *   cdrefdocum: Referencia del documento
 *   dsdocum: Descripción del documento
 *   cdusuariocrea: Código del usuario creador del documento
 *   fccrea: Fecha de creación. Formato: 15022007 13:00:12
 *   dsobserv: Observaciones
 *   cdorgacrea: Código unidad Admtiva. creadora del documento
 *   dsorgacrea: Descripción unidad Admtiva. creadora del documento
 *   cdorgaprop: Código unidad Admtiva. propietaria del documento
 *   dsorgaprop: Descripción unidad Admtiva. propietaria del documento
 *   cdorgadest: Código unidad Admtiva. destino del documento
 *   dsorgadest: Descripción unidad Admtiva. destino del documento
 *   cdextendest: Código unidad Admtiva. externa destino del documento
 *   dsextendest: Descripción unidad Admtiva. externa destino del documento
 *   cddocum_padre: Código del documento padre
 *
 */
public class EregDatosGenDocumOUT;
```

5.5. Método getInfoDocum

EregConsultaService.java

```
/**
 * Servicio para obtener el identificador de un documento a partir de su código
 * de documento.
 * Dichos datos se encuentran en la tabla SGR_DOCUMENTO
 * @param cdDocum Código del documento
 * @return Identificador del documento
 * @throws ServiceException
 */
public String getInfoDocum(String cdDocum) throws ServiceException;
```

5.6. Método getXMLAnexo

EregConsultaService.java

```
/**
 * Servicio para obtener el identificador de un documento a partir de su código
 * de documento.
 * Dichos datos se encuentran en la tabla SGR_DOCUMENTO
 * @param cdDocum Código del documento
 * @return Identificador del documento
 * @throws ServiceException
 */
public String getInfoDocum(String cdDocum) throws ServiceException;
```

5.7. Método getXMLSolicitud

EregConsultaService.java

```
/**
 * Servicio para obtener los datos de las solicitudes que los ciudadanos
 * han enviado telemáticamente.
 * Devuelve un Map<String, String> con todos los pares de campo=valor
 * de los datos del formulario.
 * @param cddocum Código de la solicitud
 * @return Map<String, String> con <campo, valor>
 * @throws ServiceException
 */
public Map<String, String> getXMLSolicitud(String cddocum)
      throws ServiceException;
```

5.8. Método loadSolicitudDocum

EregConsultaService.java

```
/**
 * Servicio que devuelve la solicitud asociada a un documento
 * @param cdDocum Código del documento
 * @return Un array de bytes con el fichero XFDF asociado al documento
 * @throws ServiceException
 */
public byte[] loadSolicitudDocum(String cdDocum) throws ServiceException;
```

6. USO DEL SERVICIO EregTramitacionService

6.1. Método bandejaEntrada4

EregTramitacionService.java

```
    /**
     * Servicio para obtener la relación de Documentos enviados a una Unidad
     * Administrativa
     * Este servicio admitirá como parámetros de entrada un rango de fechas de un
     * máximo de tres meses. Si este intervalo de fechas es superior a tres meses,
     * el sistema devolverá sólo los datos de los tres últimos meses desde la fecha
     * introducida como "fcHasta".
     * Podremos realizar la consulta por unidad o por asunto. En caso de hacerse por
     * código de asunto, también podrán pasarse como parámetros una serie de asuntos
     * separados por el carácter coma.
     * Ej: ASU01,ASU02,ASU03
     * @param datosEntrada Objeto de tipo EregBandejaEntrada4IN
     * @return Objeto de tipo EregBandejaEntrada4OUT con los datos de salida
     * @throws ServiceException
     */
    public EregBandejaEntrada4OUT bandejaEntrada4(
                EregBandejaEntrada4IN datosEntrada)
                        throws ServiceException;
```

Objeto con los datos de entrada

EregBandejaEntrada4IN.java

```
/**
 * Objeto con los datos de entrada para el método bandejaEntrada4
 * cdUnidad Unidad Administrativa
 * fcDesde Fecha desde. Formato: dd/mm/aaaa
 * situacion Situación de los Documentos en la Bandeja
 *      Pendientes = S
 *      Recibidos = N
 *      Todos = null
 *  origen Origen del documento
 *  asunto Código del Asunto
 *  itUnidadesDepen Unidades dependientes (S/N)
 *  itApoDoc Recuperar aquellos registros de entrada que sean o no de Aportación de
 *  Documentos.
 *       Aportación = S
 *       No aportación = N
 *       Todos = null
 *  cdOrden Obtener los documentos ordenados por antigüedad del Documentos
 *       Ascendente = A
 *       Descendente = D
 *
 */
public class EregBandejaEntrada4IN;
```

Objeto con los datos de salida

EregBandejaEntrada4OUT.java

```
/**
 * Objeto con los datos de salida para el método bandejaEntrada4
 * Campos:
 * numDocumentos: Número de documentos devueltos
 * documentos: Lista de documentos de tipo {@link EregDocBandejaEntrada4}
 */
public class EregBandejaEntrada4OUT;
```

Objeto EregDocBandejaEntrada4

EregDocBandejaEntrada4.java

```
/**
 * Objeto Documento de Bandeja Entrada4 que se usa en EregBandejaEntrada4OUT
 * que es el objeto de salida del método
 * Campos:
 * idDocumento: Identificador del documento
 * tipoElem: Tipo de elemento
 * idAnoReg: Identificador de la anotación registral
 * numRef: Número de referencia
 * fcEnvio: Fecha de envío
 * idOrigenDoc: Identificador del origen del documento
 * observaciones: Observaciones
 * idPropietario: Identificador del propietario del documento
 * codPropietario: Código del propietario del documento
 * descPropietario: Descripción del propietario del documento
 * tipoPropietario: Tipo de propietario del documento
 * descAsunto: Descripción del asunto
 * idEstadodoc: Estado del documento
 * fcRegEntr: Fecha-Hora de entrada en Registro
 * dsNombre: Interesado. Nombre
 * dsApellido1: Interesado. Primer apellido
 * dsApellido2: Interesado. Segundo apellido
 * dsRazonSoc: Interesado. Razón social
 * cdDocIdent: Interesado. Código de Identificación
 * dsPais: Interesado. País
 * cdProv: Interesado. Código de provincia
 * cdMuni: Interesado. Código de municipio
 * dsDireccion1: Interesado. Campo de direccion 1
 * dsDireccion2: Interesado. Campo de direccion 2
 * cdPostal: Interesado. Código Postal
 * dsTelefono: Interesado. Teléfono
 * dsFax: Interesado. Fax
 * dsEmail:  Email
 *
 */
public class EregDocBandejaEntrada4;
```

6.2. Método enviarDocum

EregTramitacionService.java

```
    /**
     * Servicio para hacer el envío de un documento en una Unidad Administrativa
     * @param cdUsuario Código usuario
     * @param cdDocum Código documento
     * @param cdUnidadorig Código unidad administrativa origen
     * @param cdUnidaddest Código unidad administrativa destino
     * @throws ServiceException
     */
    public void enviarDocum(String cdUsuario, String cdDocum, String cdUnidadorig,
          String cdUnidaddest) throws ServiceException;
```

6.3. Método recepcionarDocum

EregTramitacionService.java

```
    /**
     * Servicio para realizar la recepción de un documento en una Unidad
     * Administrativa.
     * @param cdusuario Código de usuario
     * @param cddocum Código documento
     * @param cdunidad Código de unidad
     * @throws ServiceException
     */
    public void recepcionarDocum(String cdUsuario, String cdDocum, String cdUnidad)
           throws ServiceException;
```

6.4. Método valVincDocum

EregTramitacionService.java

```
    /**
     * Servicio para validar si un documento existe, ha sido registrado de Entrada
     * en la Comunidad de Madrid, y si se precisa validar que la unidad
     * administrativa es la propietaria del mismo.
     * Devuelve el código y la descripción del Asunto, la fecha y hora de registro
     * de entrada, indicando si ha sido enviado telemáticamente o no.
     * @param cddocum Código documento
     * @param cdunidad Código Unidad Administrativa
     * @param itvalunidad ¿Validar unidad de destino? true/false
     * @return Objeto de tipo {@link EregValVincDocumOUT} con los datos de salida
     * @throws ServiceException
     */
    public EregValVincDocumOUT valVincDocum(String cddocum, String cdunidad, Boolean
validarUnidad) throws ServiceException;
```

Objeto con los datos de salida

EregValVincDocumOUT.java

```
/**
 * Objeto con los datos de salida para el método valVincDocum de
 * id_documento: identificador del documento
 * cod_asunto: Código del asunto
 * des_asunto: Descripción del asunto
 * fcRegent: Fecha y hora de Registro de Entrada varchar2(19)
 * origen: Presencial o Internet
 */
public class EregValVincDocumOUT;
```

7. ENLACES RELACIONADOS

Producto	URL
Documentación EREG	http://intranet.madrid.org/arquitecturasw/admelectronica/EREG